

# ESTRUCTURAS DE DATOS

Curso 2018/19

## PRÁCTICA 9

### Montículos

#### Instrucciones

- Se debe completar en una sesión.
- Práctica individual.
- Lee el enunciado completo antes de comenzar. Los comentarios incluidos en el código también pueden proporcionar información útil.
- Se habilitará una tarea para que entregues el código desarrollado.
- La práctica será APTA si se superan todos los test de validación proporcionados.

Vamos a utilizar montículos para implementar una cola de prioridad y el algoritmo de ordenación Heapsort. La clase **EDPriorityQueue<E>** contiene parcialmente implementada dicha cola de prioridad utilizando *minHeaps* (montículos a mínimos).

La parte privada de la clase contiene la siguiente definición:

```
E[] data;  
int size;
```

- **data**: es el vector estático donde se almacenan los elementos del montículo.
- **Size**: Cantidad de elementos válidos dentro del vector.

#### Ejercicio 1

Añade un nuevo método público a la clase **EDPriorityQueue** que permite borrar cualquier elemento **item** que reciba como parámetro:

```
T remove(T item)
```

El método devolverá el elemento borrado o **null** si no se encontraba. Recordad que el resultado tras el borrado tiene que continuar siendo un montículo. Implementa todos los métodos que consideres necesarios.

#### Ejercicio 2

Implementa un método público en la clase **EDPriorityQueue**, **void maxHeapify()**, que convierta un *minHeap* en un *maxHeap*;

### Ejercicio 3

Implementa un método público en la clase **EDPriorityQueue**, **int typeOfHeap()**, que devuelva **-1** si el vector (**data**) que almacena el montículo es un *minHeap*, **1** si es un *maxHeap* o **0** si la cola está vacía.

### Ejercicio 4

La clase **Heapsort** (en el fichero **Heapsort.java**) implementa el algoritmo de ordenación heapsort para un vector de enteros. Completa la implementación del método privado **void sink(int[] v, int p, int size)**.

Este método implementa la operación de hundir en un montículo a máximos. El *maxHeap* está almacenado en el vector **v** y tiene un tamaño **size**. El elemento que se desea hundir está en la posición **p**.